Uniform van Lambalgen's theorem fails for computable randomness

Bruno Bauwens

National Research University Higher School of Economics, Faculty of Computer Science, 11 Pokrovsky Boulevard, Kab S833, Moscow 109028, Russia.

Abstract

We show that there exists a bitsequence that is not computably random for which the odd bits are computably random and the even bits are computably random relative to the odd bits.

Keywords: Computable randomness, martingales, van Lambalgen's theorem
2000 MSC: 03D32

Imagine two referees need to decide whether a bitsequence is random. The first referee inspects the odd bits. The second referee inspects the even bits conditional to the odd ones. Both referees claim that their bits are random. Is it still possible that the full sequence is non-random? We investigate this question for computable randomness.

A *martingale f* is a function mapping strings to non-negative reals such that $f(x) = (f(x0) + f(x1))/2$ for all x. Let α and β be sequences. α is *computably random* or *CR* (respectively, *CR relative to β*) if for every totally computable martingale f (respectively, totally computable relative to β), the set of values of f on all initial segments of α is bounded.

Observe that the odd bits of a CR sequence define a sequence that is also CR. Suppose that the odd bits are CR and the even bits are CR relative to the odd ones, is the sequence CR? This question has been asked repeatedly Diamondstone et al. (2013); Miyabe and Rute (2013); Nies (2015), and we give a negative answer.

Email address: `brbauwens-at-gmail.com` (Bruno Bauwens)

Theorem 1. *There exists a non-CR sequence for which the odd bits are CR and the even bits are CR relative to the odd bits.*

This is remarkable for three reasons. First, the result is positive for the closely related notion of Schnorr randomness[1] Franklin and Stephan (2011); Miyabe and Rute (2013). (See the footnotes for details.) Secondly, with a slightly stronger assumption the answer is positive: If α is CR relative to β and β is CR relative to α, then $\alpha_1\beta_1\alpha_2\beta_2\ldots$ is CR.[2] Thirdly, it has been repeatedly claimed that a positive answer follows by the same argument as for Martin-Löf randomness. In (Miyabe and Rute, 2013, Remark 3.2) it is explained why this is not true, and Theorem 1 was conjectured.[3]

Van Lambalgen's requirement for randomness is that $\alpha_1\beta_1\alpha_2\beta_2\ldots$ is random if and only if, α is random and β is random relative to α. Both for computable and Schnorr randomness the forward implication is known to fail Yu (2007). However, for both notions of randomness, this forward direction holds when the uniform[4] variant of relative randomness is used Miyabe and Rute (2013). From Theorem 1, it follows follows that the reverse direction does not hold for computable randomness. In fact, this already follows from the weaker uniform variant, in which the even bits are only required to be uniformly random relative to the odd ones. In conclusion: The uniform van Lambalgen's criterion holds for Schnorr randomness and fails for computable randomness. This suggests that Schnorr randomness is more fundamental, and this was also observed in applications in computable analysis and reverse mathematics Rute (2015).

---

[1]A sequence α is *Schnorr random* if there exists a computable non-decreasing unbounded function h and a computable martingale d such that $d(\alpha_1\ldots\alpha_n) \geq h(n)$ for infinitely many n

[2]Indeed, any martingale is the product of a martingale that only bets on odd bits and one that bets on even bits. This decomposition can happen in a computable way. The proof finishes by a simple transformation of these martingales to conditional martingales. The converse of the statement is also true for uniformly conditional randomness, see further.

[3]Remark 3.2 considers Schnorr randomness, but is also valid for computable randomness. The conjecture is located in the paragraph below Theorem 1.3.

[4]A function f is *uniformly* computable relative to α if there exists an oracle Turing machine U such that $U^\alpha = f$ and $U^\beta(x)$ is defined for all β and x. A sequence is *uniformly random relative to α* if no martingale that is uniformly computable relative to α has unbounded values on it. We refer to Chakraborty et al. (2017) for more variants.

Proof

We use a technique that was historically introduced to prove the Gács–Kučera theorem Gács (1986); Kučera (1985). It allows to construct a sequence that encodes another sequence, and for which some martingales remain bounded, see (Downey and Hirschfeldt, 2010, Lemma 8.3.1 p325). This technique uses the following lemma.

Lemma 2. *For each martingale d, each string x and each natural number s, there exist at least two strings y of length $s+2$ such that $d(xy)/d(x) < 1+2^{-s}$.*

Proof. For all $0 < \varepsilon \leq 1/2$, at most a fraction $1/(1+\varepsilon)$ of strings y of a given length satisfy $d(xy)/d(x) \geq 1 + \varepsilon$. Hence, more than a fraction $1 - 1/(1+\varepsilon) > \varepsilon/2$ does not satisfy this property. The amount of such strings of length $s+2$ is at least $2^{s+2}\varepsilon/2$ which equals 2 for $\varepsilon = 2^{-s}$. $\qquad\square$

Proof of Theorem 1. The definition of CR-randomness does not change if we use only rational martingales (Downey and Hirschfeldt, 2010, Prop. 7.1.2 p270). We choose α to be any computably random sequence. Let $d_1, d_2, \ldots$ be an enumeration of all rational partial martingales that are partial computable with oracle α. Let $\varepsilon_s = 2^{-s}$. β is constructed (in a non-computable way) in stages, together with a total martingale d.

Initially, β is the empty string and $d = 1$. At each stage $s \geq 1$ we update β and d:

- If d_s is total relative to the oracle α and $d(\beta)$ is positive, we replace d by $d + \frac{\varepsilon_s}{d_s(\beta)}d_s$. Otherwise, d is unchanged.

- To β we append either the lexicographically first or second string y of length $s + 2$ such that

$$\frac{d(\beta y)}{d(\beta)} < 1 + \varepsilon_s \qquad (*)$$

  (by Lemma 2 there exist at least two such strings). The choice depends on whether d_{s+1} is total relative to oracle α.

- To β we again append either the lexicographically first or second string y of length $s + 2$ such that the inequality above holds. The choice now depends on the value of α_t for some fresh value t. More precisely, let t be the minimal value that exceeds $|\beta y|$, the previous value of t (for stages $s \geq 2$), and all computation times and uses of the oracle α in the evaluations of $d(z)$ for all strings z of length at most $|\beta y|$.

End of construction.

We show that β is computably random relative to α. The value $d(\beta)$ after each stage s is at most $(1+\varepsilon_1+\cdots+\varepsilon_s)\exp(2\varepsilon_1+\cdots+2\varepsilon_s)$, and this has the finite limit $2\exp 2$. Indeed, in each stage, after a possible update of d, the value $d(\beta)$ increases by ε_s, and after each extension of β, the value increases by at most a factor $1+\varepsilon_s \le \exp\varepsilon_s$.

Each test that is computable relative to α, appears in the sequence of tests. For each such test we have $d_s \le O(d)$, because at stage s, this test will be used to increment d. Hence, d_s is bounded on infinitely many initial segments of β. By the savings technique[5] this implies that no computable martingale is unbounded on β, i.e., β is computably random relative to α.

It remains to show that the pair $\alpha_1\beta_1\alpha_2\beta_2\ldots$ is not computably random. We construct a martingale e that plays only on the bits α_t that were used in the 3rd step of each stage in the construction, and on these bits the capital is doubled. In the evaluation of $e(a_1b_1\ldots a_{n-1}b_{n-1}a_n)$, we pretend that a and b are initial segments of α and β, and try to rerun the construction above. In this way, we hope to find the positions t corresponding to the bits α_t.

For this, we need to know the function d used in each stage. And for this, we need to know which functions among $d_1, d_2, \ldots$ are total. The key observation is, that knowing a prefix of β we can decide the totality of the functions $d_1, d_2, \ldots$ Indeed, knowing the updated function d in a stage s, we can compute the lexicographically first and second string for which inequality (*) holds, then observe which one is equal to the corresponding segment of β, and from this we know the totality of d_{s+1}. This allows us to update d in the next stage, and we can repeat this procedure. By choice of t, we are able to compute each index t using the prefix $\alpha_1\beta_1\ldots\alpha_{t-1}\beta_{t-1}$, and from the 3rd stage of the construction, we obtain α_t.

Detailed construction of e. Let $e(\text{empty string}) = 1$. For n-bit a and b, the values of $e(a_1b_1\ldots a_n)$ and $e(a_1b_1\ldots a_nb_n)$ are equal and are defined by induction on n. Let a be of length n and b of length $n-1$. The value of $e(a_1b_1\ldots a_{n-1}b_{n-1}a_n)$ is defined in stages. We start with $s = 1$ and $d = 1$.

At stage s we evaluate the current function d on all strings of length at most $2\sum_{i=1}^s(i+2)$. Let t be the minimal value that exceeds this length, the

---

[5]If a martingale f is unbounded on some sequence γ, then there exists a martingale that tends to infinity. For example: $\tilde{f}(x) = \sum_k 2^{-k}f(x_k)$, where x_k is equal to x if no prefix y of x satisfies $f(y) \ge 2^{2k}$, and equal to the shortest such y otherwise.

value of t in the previous stage (if $s \geq 2$) and all the computation times and oracle uses in these evaluations of d. Note that t can be infinite if one of the guesses for the totality of $d_1, \ldots, d_s$ was wrong. We only need to discriminate between the following cases:

- Case $t < |a|$. We check whether the appropriate segment of length $s+2$ of b is indeed the lexicographically first or second string y satisfying (*). (b is long enough because $|\beta y| < t \leq |a| = |b|+1$.) If this is not true, we set $e(a_1 b_1 \ldots b_{n-1} a_n) = e(a_1 b_1 \ldots b_{n-1})$, (the value of e does not matter here); otherwise, we continue the simulation and proceed to stage $s+1$ where the function d is updated according to the suggested totality of d_{s+1}.

- Case $t > |a|$. We set $e(a_1 b_1 \ldots b_{n-1} a_n) = e(a_1 b_1 \ldots b_{n-1})$.

- Otherwise we have $t = |a|$. Let i be the bit that is encoded in the 3rd step (which equals α_t for correct inputs). Let

$$e(a_1 b_1 \ldots b_{n-1} a_n) = \begin{cases} 2e(a_1 b_1 \ldots b_{n-1}) & \text{if } a_n = i \\ 0 & \text{otherwise.} \end{cases}$$

e is computable, and on initial segments of $\alpha_1 \beta_1 \alpha_2 \beta_2 \ldots$, the martingale e is unbounded. (α, β) is not CR and the theorem is proven. $\square$

Chakraborty, D., Nandakumar, S., Shukla, H., 2017. On resource-bounded versions of the van lambalgen theorem, in: International Conference on Theory and Applications of Models of Computation, Springer. pp. 129–143.

Diamondstone, D., Greenberg, N., Turetsky, D., 2013. A van Lambalgen theorem for Demuth randomness, in: Proceedings of the 12th Asian Logic Conference, pp. 115–124.

Downey, R.G., Hirschfeldt, D.R., 2010. Algorithmic Randomness and Complexity. Theory and Applications of Computability, Springer. URL: http://www.springer.com/mathematics/numerical+and+computational+mathematics/book doi:10.1007/978-0-387-68441-3.

Franklin, J., Stephan, F., 2011. Van Lambalgen's theorem and high degrees. Notre Dame Journal of Formal Logic 52, 173–185.

Gács, P., 1986. Every sequence is reducible to a random one. Information and Control 70, 186–192.

Kučera, A., 1985. Measure, Π_1^0-classes and complete extensions of PA, in: Recursion theory week. Springer, pp. 245–259.

Miyabe, K., Rute, J., 2013. Van Lambalgens theorem for uniformly relative Schnorr and computable randomness, in: Proceedings of the 12th Asian Logic Conference, World Scientific. pp. 251–270.

Nies, A., 2015. Logic blog 2014. Preprint arXiv:1504.08163 .

Rute, J., 2015. New directions in randomness. Slides http://math.uni-heidelberg.de/logic/conferences/ccr2015/.

Yu, L., 2007. When van Lambalgen's theorem fails. Proceedings of the American Mathematical Society , 861–864.